IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant: Thomas GOERING

Appl'n No.: 10/665,305

Filing Date: 22 September 2003

For: System and Method for Reusing Form Elements in a Form Building Application

Confirmation No. 5450

Group Art Unit: 2176

Examiner: R.S. Desai

Mail Stop APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF UNDER 37 C.F.R. 41.41

Dear Sir:

In response to the Examiner's Answer mailed on April 3, 2008, for the above-identified application, submitted herewith include:

- **STATUS OF CLAIMS**, which begins on page 2;
- **GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL** which begins on page 2; and
- **ARGUMENT** which beings on page 4.

I. STATUS OF CLAIMS

Pending claims 1-18 stand finally rejected and are the subject of this appeal. Claim 19 is canceled.

II. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-18 are unpatentable under 35 U.S.C. §103(a) over U.S. Patent Application Publication No. 2005/0080756 A1 to Hitchcock (hereinafter "Hitchcock").

III. ARGUMENT

The Final Rejection and Examiner's Answer fail to demonstrate that Hitchcock renders obvious any of pending claims 1-18. Specifically, Hitchcock does not teach or suggest invalidating output modules or regenerating invalidated output modules as recited in independent claims 1, 9, 17. In our Appeal Brief, we explained that Hitchcock's disclosure is limited to changing data records but this disclosure is insufficient to reach processes that invalidate output modules and regenerate previously invalidated output modules. As recited in the pending claims, "output modules" are executable modules of a software system.

The Examiner's Answer argues that Hitchcock's form data is the same as an output module but, at the same time, impliedly acknowledges they are not (Examiner's Answer, lns. 8-9). In essence, the Examiner's analysis concludes that the pending claims cannot be patentable because he believes the two systems can be used to achieve a common result and the structural differences among the claims – output modules vs. form data – cannot be given patentable effect. The Examiner's Analysis is wrong, of course. An obviousness analysis does not permit a result-oriented analysis such as this. Instead, one must determine that the prior art discloses all elements of the pending claims before the claims can be rejected as obvious. 35 U.S.C. § 103. The Examiner has not demonstrated that the cited art discloses all elements of the claims in this case.

A. Hitchcock's Current Form Does Not Correspond to an "Output Module" as Claimed

The Examiner's Answer equates the output modules presented in the pending claims with data records of Hitchcock (called the "current form"). Examiner's Answer, p. 14, lns. 2-3. This analysis reflects incorrect interpretation of the claims and art and also a result-oriented analysis which is inappropriate for a prior art rejection.

1. The Examiner's Overlooks Differences between Output Modules and Form Data.

As claimed, an output module is an executable module of a software system. This construction flows naturally from the ordinary meaning of the term "module" as used in software fields. See, Webopedia, for example, "a module is a part of a program.[1]" This construction also is consistent with the Specification:

> Once a form is created, it may be activated by clicking activation button, upon which the Smart Forms runtime system checks the form and automatically generates an ABAP function module (i.e., form *output module*) that can subsequently be called by an application program, for example, to create delivery notes in Sales and Distribution. Specification, para. 6.

Thus, the claimed "output module" is an executable module.

The Examiner is incorrect to equate Hitchcock's current form with the output modules as claimed. Hitchcock does not disclose that the current forms are executable modules. They are merely data records. Accordingly, the Examiner's rejection cannot be sustained.

2. The Examiner Fails to Recognize that Hitchcock Discloses Executable Modules – They Never Change.

Interestingly Hitchcock does disclose elements of his system that are executable modules. For example, Hitchcock's system includes a single forms engine program that displays information from an application database in the format prescribed by the application data file. Hitchcock, para. 64. Hitchcock has no disclosure to suggest that the forms engine program is invalidated or regenerated. Indeed, Hitchcock states the "applicant database can be extended to include new attributes ***without making any changes to the forms engine program***". Hitchcock, para. 65, lns. 1-3 (emphasis added). Hitchcock, therefore, teaches to use a single engine program continuously throughout operation, which stands in stark contrasts to the invalidation and regeneration processes claimed here.

[1] http://www.webopedia.com/TERM /m/module.html last accessed June 3, 2008

These differences were identified to the Examiner in prior correspondence and in our Appeal Brief at page 4 last paragraph. In response to this argument, the Examiner states:

> Examiner's position indicated that although Hitchcock does not expressly state the output module is invalidated, he does teach that as the user enters or customizes data, the data is shared through an extensible database between disparate forms.

Examiner's Answer, p. 14, lns. 6-9. Of course, as discussed above in part A, data records are not output modules. They are not executable content. They cannot be called by other applications as disclosed in the Specification, para. 6. The Examiner is wrong to equate them in this manner.

Hitchcock teaches away from the claimed invention, because Hitchcock expressly teaches that all the form changes are done by changing information in the database ("application description file can be easily modified") and not by invalidating an executable module, the "output modules." Hitchcock, para. 64.

3. The Examiner Defies Logic to Assert that Changed Data Corresponds to Invalidated and Regenerated Output Modules.

To further confuse the differences between Hitchcock's forms and the claimed output modules, the Examiner alleges that Hitchcock invalidates a data object when any change occurs to stored data. The Examiner states:

> If an output module is responsible for outputting data for presentation and the data has been altered in a manner that affects data in other forms, then the output module is "invalided."

Examiner's Answer, p. 15, lns. 10-12. Again this construction depends on a flawed interpretation. on construing "output module" to the current form or the currently displayed form. In other words, the Examiner is stating that if some of the data that an "output module" that includes executable code changes then the executable code is invalidated, because the executable code will now create a different display. *Ibid.* This is simply not a tenable argument. The Hitchcock system is very similar to the prior art described in the Specification at paragraphs 7 and 8 . A change in form data does not necessitate a change in Hitchcock because the form data is referenced in Hitchcock by reference and stored in a database.

B. One of the Steps in the Claimed Method is Not Abstract and Should be Given Patentable Weight.

The Examiner argues that the "invalidating" step is abstract, Examiner's Answer, p. 14. This analysis is wholly incorrect; modern software systems use a variety of techniques to identify which modules are eligible to be called and used by other software components. For example, the Specification discloses using flags to indicate which modules are valid and which are not:

> As shown in FIG. 9, when the incorporated reusable form element is later changed (step 900), ***form manager 525 invalidates all applicable output modules (step 910) by changing their valid flag 610 entries to FALSE***.

Specification, para. 43. Note that these flags permit the invalid output modules to persist in storage but they are invalid for use due to the state of their valid flags. Thus, as disclosed and claimed, the invalidating step defines an affirmative step to be taken during performance of the claimed method.

The Examiner argues that the "invalidating" step is abstract because it does not require there to be a "tangible" indication that an output module is invalid. This is off point. The claim refers to a control method that selectively invalidates output modules and reinstates them as described in the claims. The claims refer to management and control of output modules which defines affirmative acts.

The invalidating step defines an affirmative step of the pending claims. To reject claims 1, 9 or 17, the Examiner must provide prior art that discloses an invalidation of an output module. As discussed above, Hitchcock's forms are not output modules. Additionally, Hitchcock's disclosed processes for changing an element of data does not teach or suggest any process for ***invalidating*** output modules or for ***regenerating*** invalidated output modules. The Examiner's analysis is wrong.

C. A Form Element as Claimed is not Form Data.

The Examiner's Answer argues:

> In other words, as stated in the paragraph above, a change to a *data element* does affect the output module in that it invalidates the module because the data element has been altered.

Examiner's Answer, p. 16, lns. 1-3, emphasis added. (The Examiner apparently mistakenly wrote "data element" for "form element.") The Examiner is then construing a "form element" as claimed to be the form data of Hitchcock. But, the Examiner's argument completely ignores the claimed invention which regenerates the output module based on a form element being changed. The "form element" is generated into the "output module" as is explained in paragraph 8 of the Specification. Construing "form element" as "form data" also depends on construing the output module as a displayed form, which is improper as discussed above.

D. Obviousness Rejections Should not be Based on the So-Called 'Validity' of the Specification.

In justifying the outstanding obviousness rejections, the Examiner's Answer questions, without analysis, the 'validity' of paragraph 8. Examiner's Answer, p. 16, lns. 3-5. This is an inappropriate consideration for obviousness analyses. To determine whether a claim is obvious, one must compare the claims against the scope and content of the prior art. Although claims are to be interpreted in light of the specification, the recitations of the claims must be compared to the art. Claims are held valid or invalid, not specification paragraphs. The Examiner is incorrect to base his obviousness analysis on specification description.

Appellants note there are no outstanding 35 U.S.C. § 112 rejections, not for enablement, written description or best mode. It is legal error to consider the so-called 'validity' of a paragraph from an applicant's disclosure in an attempt to justify a rejection based on prior art.

IV. CONCLUSION

Applicant respectfully requests that the Board of Patent Appeals and Interferences reverse the Examiner's decision rejecting claims 1-18 and direct the Examiner to pass the case to issue. These claims are allowable over the cited art.

Respectfully submitted,

/ Gregory R. Grace /

Date: June 3, 2008

Gregory R. Grace
Registration No. 59,733

KENYON & KENYON LLP
1500 K Street, N.W., Suite 700
Washington, D.C. 20005
(202) 220 - 4200 (telephone)
(202) 220 - 4201 (facsimile)